State of Utah

Web Standards: Part 5.0 Mobile Platform Design Guidelines for Touch Enabled Devices

Department of Technology Services

January 31, 2011

State of Utah Web Standards and Guidelines: Part 5.0 Mobile Platform Design Guidelines for Touch Enabled Devices

Department of Technology Services
Office of the Chief Technology Officer
January 31, 2011
DTS Technology Standard 4300-0001-05

Status: TBD
Effective Date: March 2011
Next Review Date: March 2012
Approved By: Stephen Fletcher, CIO

Legal and Statutory Authority

Authority

The Department of Technology services is charged in *Utah Code 63F-1-104 et seq* with the overall responsibility for defining technology standards. The Chief Information Officer (CIO) has rulemaking and policy making authority for technology standards and practices for the Executive Branch agencies as specified in Utah Code 63F-1-206 et seq, with the exception of those agencies exempted in Utah Code 63F-1-102.

Implementation

Application of these standards, recommendations, and guidelines shall be applied to all new Mobile Web site development in the State of Utah commencing March 2011 as advisory guidelines and recommendations. Existing mobile applications and sites should be made compliant as the sites are redesigned. Recommended practices and guidelines should be incorporated on a best effort basis.

Changes and Revisions

Suggestions for changes, revisions, and additions to this document are encouraged. This standards document will be subject to review and revision one year from the initial date of publication, and every twelve months thereafter. Change recommendations will follow approved Architecture Review Board processes for enterprise standards approval and changes.

1.0 Introduction

The recommendations in this guideline are intended to improve the experience of the Web on mobile devices. The recommendations are made in the context of working towards a One Web experience. One Web as defined by the W3C, “means making, as far as is reasonable, the same information and services available to users irrespective of the device they are using. However, it does not mean that exactly the same information is available in exactly the same representation across all devices.” These guidelines specifically reference best practices and approaches that will enable State Websites to work effectively on a wide range of touch based smart phones and tablets with varying screen sizes and capabilities.

1.1 Purpose

This guideline identifies mobile device platform requirements, considerations, and best practices with links to relevant W3C and vendor human interface guidelines. The intent of the document is to encourage Web developers to look at design from a multi-platform perspective for all State Websites.

1.2 Audience

The primary audience for this document are agency business leaders and Web development staff that want to be sure that agency Web resources are cross platform and effective for end users.

1.3 Scope

This document identifies considerations and guidelines from a business perspective with links to more technical implementation details and practices.

2.0 Requirements

This statement of requirements is illustrative and is not intended to be exhaustive or complete. Requirements from this section are adapted and taken from the W3C Mobile Web Best Practices guideline document and other vendor supplied human interface guidelines.

From a high level perspective agency Web sites need to be designed so they perform usefully on mobile devices. Several factors need to be considered in site design and implementation:

2.1 Testing. Plan on the site working in mobile browsers and test the site on the targeted browser and platforms as a component of existing browser site testing practices.

2.2 Device Detection. Use user agent detection code to avoid sending mobile

device browsers the wrong version of your site.

2.3 Designing for Diverse Platforms. Use Page and navigation design approaches that do not crowd elements too closely together for touch interaction.

2.4 Coding and Site Implementation.

- Use W3C standard web technologies instead of plug-ins.
- Check viewport tag settings to ensure that width is compatible with targeted mobile device browsers.
- Modify code that relies on CSS fixed positioning.
- Prepare for a touch interface, avoid use of mousemove, mouseover, mouseout or the CSS pseudo-class :hover. Use browser supported touch events.
- Use textareas instead of contenteditable elements.

3.0 Guidelines

The guidelines discussed in this section tie directly to the requirements identified is section 2. Additional detailed specifications are available in section 4 Best Practices, and section 5 References.

3.1 Testing

Many websites perform server-side checks on the browser's user agent string to determine whether or not they should serve the mobile version of their website. Tablet devices such as the iPad are capable of delivering a "desktop" web experience, and users will expect this experience since iPad has a large screen and fast network connectivity. If you have a version of your website that is optimized for mobile devices with small screens, do NOT serve this mobile version to iPad or other tablet users. As of the close of 2010 approximately 85% of smart phone and tablet devices used webkit based browser platforms making this the de-facto testing standard for these mobile devices.

> ***Recommendation:*** Agency Websites should be tested on targeted mobile platforms to detect issues that may need to be corrected.

3.2 Device Detection

There are three options for handling site presentation with user agent detection or device capability detection for mobile devices:

1. Follow cross-device standards and keep your service as simple as possible to ensure it performs on all possible devices.
2. Offer two optimized services—one for large screens; the other for small screens—then detect device capabilities and route the user to the proper stylesheet for the device.

3. Detect the type of device using media queries and/or user-agents, infer each device's characteristics, then offer an optimized service for each device.

   Recommendation: Option 3. Use user-agents as needed for initial device profiling. Detect the type of device and use media queries for templating to detect the mobile device and serve the content appropriately for that device.

   Utah.gov uses the media query method as follows for the iPad:

   ```
   <!-- iPad Stylesheet -->
   <link rel="stylesheet" media="only screen and (max-device-width: 1024px)" h    ref="css/ipad.css" type="text/css" />
   ```

   The example that follows uses media query to detects orientation and device size for the iPhone and iPad in three stylesheets:

   ```
   <link rel="stylesheet" media="all and (max-device-width: 480px)" href="iphone.css">
   <link rel="stylesheet" media="all and (min-device-width: 481px) and (max-device-width: 1024px) and (orientation:portrait)" href="ipad-portrait.css">
   <link rel="stylesheet" media="all and (min-device-width: 481px) and (max-device-width: 1024px) and (orientation:landscape)" href="ipad-landscape.css">
   <link rel="stylesheet" media="all and (min-device-width: 1025px)" href="ipad-landscape.css">
   ```

   Different devices use different screen and max device width settings to deliver content to the device, but media queries is applicable to all remote devices. Using media queries is best when agencies have developed standard CSS templates.

3.3 Designing for Diverse Platforms

There are a number of design considerations that apply to content delivery for mobile devices:

- Design for smaller and varying screen areas, and avoid congested displays.
- Design user interfaces that are optimal for mobile devices.
- Use mobile device stylesheets.
- Apply W3C Mobile Device Best Practices.
- Leverage device capabilities, and orientation options.
- Layout for effective interaction with touch.

- Be sure mobile implementations meet agency goals and user needs.
- Be sure the mobile version of the site provides access to essential services.

3.4 Coding and Site Implementation
Use W3C standard technologies. Plug-ins are not supported on most mobile Webkit browser implementations.

If your current site uses a plug-in to display user interface elements such as menus or other navigation elements, these elements will not be accessible to iOS and most Android device users. Include a code path for platforms that do not support plug-ins, and for users on desktop platforms such as Mac OS X or Windows that may have plug-ins disabled.

The following best practices W3C web technologies for mobile devices have been identified and are listed here for convenience.

1. [THEMATIC_CONSISTENCY] Ensure that content provided by accessing a URI yields a thematically coherent experience when accessed from different devices.
2. [CAPABILITIES] Exploit device capabilities to provide an enhanced user experience.
3. [DEFICIENCIES] Take reasonable steps to work around deficient implementations.
4. [TESTING] Carry out testing on actual devices as well as emulators.
5. [URIS] Keep the URIs of site entry points short.
6. [NAVBAR] Provide only minimal navigation at the top of the page.
7. [BALANCE] Take into account the trade-off between having too many links on a page and asking the user to follow too many links to reach what they are looking for.
8. [NAVIGATION] Provide consistent navigation mechanisms.
9. [ACCESS_KEYS] Assign access keys to links in navigational menus and frequently accessed functionality.
10. [LINK_TARGET_ID] Clearly identify the target of each link.
11. [LINK_TARGET_FORMAT] Note the target file's format unless you know the device supports it.
12. [IMAGE_MAPS] Do not use image maps unless you know the device supports them effectively.
13. [POP_UPS] Do not cause pop-ups or other windows to appear and do not change the current window without informing the user.

14. [AUTO_REFRESH] Do not create periodically auto-refreshing pages, unless you have informed the user and provided a means of stopping it.
15. [REDIRECTION] Do not use markup to redirect pages automatically. Instead, configure the server to perform redirects by means of HTTP 3xx codes.
16. [EXTERNAL_RESOURCES] Keep the number of externally linked resources to a minimum.
17. [SUITABLE] Ensure that content is suitable for use in a mobile context.
18. [CLARITY] Use clear and simple language.
19. [LIMITED] Limit content to what the user has requested.
20. [PAGE_SIZE_USABLE] Divide pages into usable but limited size portions.
21. [PAGE_SIZE_LIMIT] Ensure that the overall size of page is appropriate to the memory limitations of the device.
22. [SCROLLING] Limit scrolling to one direction, unless secondary scrolling cannot be avoided.
23. [CENTRAL_MEANING] Ensure that material that is central to the meaning of the page precedes material that is not.
24. [GRAPHICS_FOR_SPACING] Do not use graphics for spacing.
25. [LARGE_GRAPHICS] Do not use images that cannot be rendered by the device. Avoid large or high resolution images except where critical information would otherwise be lost.
26. [USE_OF_COLOR] Ensure that information conveyed with color is also available without color.
27. [COLOR_CONTRAST] Ensure that foreground and background color combinations provide sufficient contrast.
28. [BACKGROUND_IMAGE_READABILITY] When using background images make sure that content remains readable on the device.
29. [PAGE_TITLE] Provide a short but descriptive page title.
30. [NO_FRAMES] Do not use frames.
31. [STRUCTURE] Use features of the markup language to indicate logical document structure.
32. [TABLES_SUPPORT] Do not use tables unless the device is known to support them.
33. [TABLES_NESTED] Do not use nested tables.
34. [TABLES_LAYOUT] Do not use tables for layout.

35. [TABLES_ALTERNATIVES] Where possible, use an alternative to tabular presentation.
36. [NON-TEXT_ALTERNATIVES] Provide a text equivalent for every non-text element.
37. [OBJECTS_OR_SCRIPT] Do not rely on embedded objects or script.
38. [IMAGES_SPECIFY_SIZE] Specify the size of images in markup, if they have an intrinsic size.
39. [IMAGES_RESIZING] Resize images at the server, if they have an intrinsic size.
40. [VALID_MARKUP] Create documents that validate to published formal grammars.
41. [MEASURES] Do not use pixel measures and do not use absolute units in markup language attribute values and style sheet property values.
42. [STYLE_SHEETS_USE] Use style sheets to control layout and presentation, unless the device is known not to support them.
43. [STYLE_SHEETS_SUPPORT] Organize documents so that if necessary they may be read without style sheets.
44. [STYLE_SHEETS_SIZE] Keep style sheets small.
45. [MINIMIZE] Use terse, efficient markup.
46. [CONTENT_FORMAT_SUPPORT] Send content in a format that is known to be supported by the device.
47. [CONTENT_FORMAT_PREFERRED] Where possible, send content in a preferred format.
48. [CHARACTER_ENCODING_SUPPORT] Ensure that content is encoded using a character encoding that is known to be supported by the device.
49. [CHARACTER_ENCODING_USE] Indicate in the response the character encoding being used.
50. [ERROR_MESSAGES] Provide informative error messages and a means of navigating away from an error message back to useful information.
51. [COOKIES] Do not rely on cookies being available.
52. [CACHING] Provide caching information in HTTP responses.
53. [FONTS] Do not rely on support of font related styling.
54. [MINIMIZE_KEYSTROKES] Keep the number of keystrokes to a minimum.
55. [AVOID_FREE_TEXT] Avoid free text entry where possible.

56. [PROVIDE_DEFAULTS] Provide pre-selected default values where possible.
57. [DEFAULT_INPUT_MODE] Specify a default text entry mode, language and/or input format, if the device is known to support it.
58. [TAB_ORDER] Create a logical order through links, form controls and objects.
59. [CONTROL_LABELLING] Label all form controls appropriately and explicitly associate labels with form controls.
60. [CONTROL_POSITION] Position labels so they lay out properly in relation to the form controls they refer to.

    Recommendation: Implement W3C standard technologies as options to plugin technologies used on desktop site implementations, and serve these options based upon browser and device characteristics.

3.4.1 Check viewport tag settings to ensure that width is compatible with targeted mobile device browsers. If you have specified viewport settings for your webpage, verify that these same settings are suitable for other devices such as the iPad .

Specifically, if you want the width of the viewport to match the width of the device, you should use the device-width constant instead of a hard-coded pixel value.

> ***Recommendation:*** Use a constant for viewport width.
> <meta name="viewport" content="width=device-width" />

3.4.2 Modify code that relies on CSS fixed positioning.

On a mobile device the viewport is analogous to the window in a desktop browser. While elements that use fixed positioning in Safari, for example in OS X and Windows always stay on screen, elements that use fixed positioning in mobile devices can end up offscreen as users zoom and pan the webpage.

Safari on iPad and Safari on iPhone, and most other Webkit browsers do not have resizable windows. The window size is set to the size of the screen (minus browser user interface controls), and cannot be changed by the user. To move around a webpage, the user changes the zoom level and position of the viewport as they double tap or pinch to zoom in or out, or by touching and dragging to pan the page. As a user changes the zoom level and position of the viewport he/she are doing so within a viewable content area of fixed size (that is, the window). This means that webpage elements that have their position "fixed" to the viewport can end up outside the viewable content area, offscreen.

Recommendation: Do not use page presentation code that relies on CSS fixed positioning on mobile devices.

3.4.3 Prepare for a touch interface, avoid use of mousemove, mouseover, mouseout or the CSS pseudo-class :hover. Use vendor browser supported touch events.

The examples that follow have been adapted from Technical Note TN2262: Preparing Your Web Content for iPad. The concepts in these examples are applicable to other mobile device platforms.

Although an external hardware keyboard is an option for use with iPad and some other vendor mobile devices, the primary means of interacting with web content is through touch. On the iPad and iPhone, the software keyboard appears when a form control that requires text input — such as <input type="text"> or <textarea> — gains focus. Users should not be forced to rely on a keyboard to navigate the webpage.

Users interact with your web content directly with their fingers, rather than using a mouse. This creates new opportunities for touch-enabled interfaces, but does not work well with hover states. For example, a mouse pointer can hover over a webpage element and trigger an event; a finger on a Multi-Touch screen cannot. For this reason, mouse events are emulated. As a result, elements that rely only on mousemove, mouseover, mouseout or the CSS pseudo-class :hover may not always behave as expected on a touch-screen device such as an iPad or iPhone.

Touches can be handled directly or advanced gestures can be detected, using the DOM Touch events touchstart, touchmove, touchend, and touchcancel. Unlike mouse events which are emulated, DOM Touch events are specifically designed to work with touch interfaces, so their behavior is reliable and expected. For more information on using touch events in web content see vendor specific guides such as the "Handling Events" section of the Safari Web Content Guide, the Touch, TouchEvent, and TouchList classes in the Safari DOM Extensions Reference, and the SlideMe sample code at the Safari Dev Center. Other vendors offer similar resources.

Since touching and holding in Safari on iPhone OS will invoke the Cut/Copy/Paste dialog, you may also choose to disable selection on user interface elements such as menus and buttons using -webkit-user-select: none. It is important to only disable selection as needed on a per-element basis. Selection in webpages should never be globally disabled.

Recommendation: Implement non-hover methodologies for mobile site implementations. Use touch event handling capabilities when available.

3.4.4 Use textareas instead of contenteditable elements.

contenteditable is not supported on Webkit based mobile browsers such as Safari. If you are using contenteditable to enable text input within a styled element

(for instance, <p contenteditable> or <div contenteditable>), you can replace this styled element with a styled <textarea>.

In Safari on iPad, iPhone, Mac OS X, and Windows, you can customize the appearance of <textarea> elements using CSS. If necessary, you can even disable any platform-specific, built-in styling on a <textarea> by specifying -webkit-appearance: none.

> ***Recommendation:*** Use textareas to enable text input within a styled element.

5.0 References

- Apple Inc. iPad Human Interface Guidelines. iOS Reference Library,September 8, 2010.
- Apple Inc. iPhone Human Interface Guidelines. iOS Reference Library, September 1, 2010.
- Apple, Inc. Technical Note TN2262: Preparing Your Web Content for iPad, March 2010.
- Google. User Interface Guidelines. Android Developers, October 12, 2010. Design guidelines for Android.
- Microsoft. Designing Applications for Windows Mobile Platforms. MSDN, April 19, 2010. Design guidelines for Windows Mobile 6.5.
- Nokia. Design and User Experience Library. Version 2.0. Nokia Forum, June 16, 2010.
- Palm, Inc. User Interface Guidelines. Palm Developer Center, 2010. Design guidelines for the webOS 2.0 software platform for the Palm Pre.

- Research in Motion. Designing Applications for BlackBerry Devices. Version 6.0. BlackBerry, 2010.
- _____. BlackBerry Smartphones: UI Guidelines. Version 6.0. Waterloo, Ontario, Canada: Research in Motion, 2010.
- Robson, Elizabeth, *Using CSS Media Queries to Style Your iPhone and iPad HTML*, O'Reilly Community, April 17, 2010
- Six, Janet M., Going Mobile: Designing for Different Screen Sizes | Promoting Your Mobile App, UX Matters, October 18, 2010.
- _____, Going Mobile, Part II: When to Go Mobile | Reuse Your Web Design or Start from Scratch? UX Matters, November 8, 2010.
- W3C. Media Queries: W3C Candidate Recommendation, July 27, 2010
- ____, Mobile Web Best Practices.Version 1.0. W3C, July 29, 2008. Design guidelines for mobile Web applications."

Comments

Please direct comments and suggestions on this document to
Robert Woolley via e-mail at bwoolley@utah.gov or by phone at (801) 538-1072.